Abstract

Darknet markets provide a large platform for trading illicit goods and services due to their anonymity. Learning an invariant representation of each user based on their posts on different markets makes it easy to aggregate user information across different platforms, which helps identify anonymous users. Traditional user representation methods mainly rely on modeling the text information of posts and cannot capture the temporal content and the forum interaction of posts. While recent works mainly use CNN to model the text information of posts, failing to effectively model posts whose length changes frequently in an episode. To address the above problems, we propose a model named URM4DMU(User Representation Model for Darknet Markets Users) which mainly improves the post representation by augmenting convolutional operators and self-attention with an adaptive gate mechanism. It performs much better when combined with the temporal content and the forum interaction of posts. We demonstrate the effectiveness of URM4DMU on four darknet markets. The average improvements on MRR value and Recall@10 are 22.5% and 25.5% over the state-of-the-art method respectively.

Index Terms— Darknet Markets, User Representation, Self-attention Mechanisms, Convolution Networks, User Behaviors

1. Introduction

Darknet markets contain vast resources for the illicit drug trade, adult content and other illicit services~~, and the~~. The anonymity of the darknet makes it an ideal environment for~~crime~~ criminal discussions. Users on darknet markets are resilient to closures. They will ~~rapidly~~ quickly migrate to newer markets ~~occur~~ when one market shuts down. Learning an invariant representation for each user based on their posts on different markets will link malicious users ~~using~~ who use multiple accounts, which makes it convenient to aggregate user information across different platforms. It provides more useful information for the analysis of anonymous users' identities, making user representation learning on these markets become a compelling problem.

Traditional techniques for such tasks rely upon feature engineers, that is, using features from text corpora, such as the high-frequency words, capitalization, punctuation style, word or character n-grams and function words usage, as a user's 'signature'. However, such techniques perform poorly for short text corpora in highly anonymized environments. Convolutional neural networks (CNN) are introduced for modeling the text in user representation model. On this basis, Andrews and Witteveen (2019) proposed to combine the analysis of graph context information and temporal characteristics with text representation for user representation. Pranav Maneriker (2021) developed a novel stylometry-based multitask learning approach that leverages graph context to construct low-dimensional representations of short episodes of user activity on darknet markets. They first applied the method to the analysis of dark web data and achieved good results on four darknet market forums datasets. However, the core component of the method mainly relies upon the convolution operation. It has a significant weakness ~~in~~ that it only operates on a local neighborhood, thus missing global information. Transformer has emerged as a recent advance to capture long-range interactions and has mostly been applied to sequence modeling. However, since the transformer takes into account all the elements with a weighted averaging operation that disperses the attention distribution, it may overlook the relation of neighboring elements (i.e. n-grams) ~~that~~ which are

important for modeling text.

As shown in Fig1, the length of a user's posts may change in a large range frequently, ~~it~~ which requires the model to be able to not only ~~can~~ capture the local features but also ~~can~~ understand the long-range post content. Therefore, we propose to concatenate convolutional feature maps with a set of feature maps produced via self-attention through an adaptive gate mechanism to address the above challenges. In summary, our main contributions are as follows:

First, we proposed a user representation model for darknet markets users URM4DM. It takes advantage of the CNN in capturing local features and the transformers in modeling long sequential information and ~~combining~~ combines the temporal content and forum interactions of the user. It is more adaptive to model posts with various lengths of the same user in comparison with traditional methods.

Second, we reveal the importance and remarkable effect of combining the temporal content and forum interactions with our text representation method for modeling darknet markets users.

Third, we demonstrate the effectiveness of URM4DRU on four darknet markets – Black Market Reloaded, Agora Marketplace, Silk Road, and Silk Road 2.0. The average improvements on MRR value and Recall@10 are 22.5% and 25.5% over the state-of-the-art method respectively, which highlights the benefits of URM4DMU.

2. METHOD

As shown in Fig 2, URM4DMU contains two key components, namely the post embedding and the episode embedding. The post embedding is derived from the text, time and the structural context of a post. We further use the episode embedding to model users' posts over a period on darknet markets. Each episode e of length Le consists of multiple tuples of texts, times, and contexts $e = \{(ti, \tau i, mi) \mid \mid 1 \leqslant i \leqslant Le\}$. A neural network architecture $f(\theta)$ maps each episode e to combined representation $e \in RE$. We design ~~the~~ a metric learning task to ensure that episodes ~~having~~ with the same author have similar representations.

2.1 Text Embedding

The text embedding is designed to extract semantic features from the post and project the extracted semantic features into the information space. The sentence of the post is first padded to maintain the uniform length Lp of all sentences. Each token of the sentence is mapped to a dt dimensional continuous space by a one-hot coding layer followed by an embedding matrix Et of dimensions $|V| * dt$ where V is the token vocabulary. The post is represented as a sequence embedding of dimension $Lp * dt$ which is concatenated by all the token vectors. Then the model input is $X = [x0, x1, \cdots, xLp-1]$. After that, we apply the CNN to extract local features, the transformer to model the long sequential information, and a valve component to learn the importance of the above two components for user modeling ~~the users~~.

2.1.1 Convolutional Over Text

The CNN is to extract local features of the token sequence. The multiple convolution kernels with different sizes (sizes are 2,3,4,5) are set to extract key information and capture local correlations:

Formula(1)

C: is fed into a dense layer:

H is the output of CNN which involves key local features, ~~the~~ The dimensional is d_c.

2.1.2 Self-attention Over Text

After getting the sequence representation X, the Positional Encoding is introduced to get position information. The vector representation of the token will be obtained by adding both of them. Then, three vectors qt, kt, vt are obtained according to the embedding vector xt($0 \leqslant t \leqslant Lp - 1$). A score for each vector is computed with the following equation: score = qt × kt. To stabilize the gradient, ~~the~~ score normalization is used. It is divided by $\sqrt{dk}$. The score value goes through the Softmax activation function to obtain the weight. Once the weights ~~is~~ are obtained, they are multiplied by the value vectors of the corresponding tokens vt, and a weighted score for each input vector vt is achieved. The final output is $z = \sum_{i=0}^{Lp-1} vt$. The calculation formula is as follows:

Formula(2)

Multi-Head Attention (MHA) can provide m representation subspaces for attention. In each subspace, different Q, K, V weight matrices are trained, and each matrix is randomly initialized and generated. The token embeddings are then trained to project into different representation subspaces.

Formula(3)

Then the latent semantic feature map S is produced through Feed Forward and Add&Norm, π is the number of MHA:

Formula(4)

2.1.3 Valve Component

To take advantage of the CNN in capturing local features and the transformers in modeling long sequential information, we apply pooling layers and dense layers to project S into the ds dimensional information space:

Formula(5)

We fuse HS max, HS mean and HC to output a dt($dt = dc + 2ds$) dimensional information-enhanced semantic feature map HO through the AdaGate function:

Formula(6)

where $\odot$ stands for an element-wise product. The values in $H\lambda$ are in probability form, and the Valve function is designed to restore less-confident entries (with probability near 0.5) for matching with elements in (HS max + HS mean). Concretely, for every unit $\lambda \in H\lambda$

Formula(7)

where ε is an empirical hyper-parameter that is used for tuning the threshold of confidence. Specifically, we dump all global information if $\varepsilon = 0$, and accept all global information if $\varepsilon = 0.5$. Therefore, the element-wise production exploits Valve as a filter that only extracts necessary information.

2.2. Time Embedding

The temporal content plays a key role in modeling the user [10, 11], especially for our model. It contains important information ~~that~~ indicating when the post was created and is available at different granularities across darknet market forums. It involves some behavior of users on the darknet market. To obtain a consistent time embedding, we only consider all the date information (date) with the smallest granularity available in the market. The days of the week and date (i.e. day of the year) ~~for~~ of each post are used to compute the time embedding by projecting it into

a d_τ dimensional vector HT as a global time stamp. It ~~was~~ is fed into the time encoder for time embedding.

2.3. Structural Context Embedding

The structural context embedding aims at capturing user behavior on forums, which is first proposed by previous work [11]. A heterogeneous graph is constructed from forum posts by four types of nodes: user (U), sub-forum (S), thread (T), and post (P)~~, and each~~. Each edge indicates either a post of a new thread (U-T), reply to an existing post (U-P) or an inclusion (T-P, S-T) relationship. The metapath2vec framework [18] with specific meta-path schemes is designed for darknet forums to capture user behavior. To fully capture the semantic relationships in the heterogeneous graph, all meta-paths starting from and ending at a user node are considered. Seven meta-path schemes: UPTSTPU, UTSTPU, UPTSTU, UTSTU, UPTPU, UPTU, and UTPU are designed to capture user behavior. A final d_m dimensional embedding HM is generated for the context of a post. The learned embeddings will preserve the semantic relationships between each subforum, ~~included~~ including posts and relevant users.

2.4. Episode Embedding

The embeddings of each component of a post in an episode are concatenated into a $d_e = d_t + d_\tau + d_m$ dimensional embedding. An episode with L posts has a $L * d_e$ embeddings. We follow the architecture proposed by Andrews and Bishop (2019) [10]. After episode embedding, we can use a final metric learning loss corresponding to the task-specific $g(\phi)$, and then train the parameters θ and ϕ. The framework, as mentioned above, results in a ~~model trainable~~ trainable model referred to as Single-Task Learning for a single market $Market_i$. Note that the first half of the framework (i.e., $f(\theta)$) is sufficient to generate embeddings for episodes, making the module invariant to the choice of $g(\phi)$. However, the embedding modules learned from these embeddings may not be compatible with comparisons across different markets, which motivates our multi-task setup.

Multi-Task Learning. Same as the architecture proposed by Maneriker et al, we use the cross-dataset to help train the model by combining four different datasets. Since then, the task-specific metric learning layer g_ϕ Taski is selected and a task-specific loss is backpropagated through the network. Note that in the ~~cross-dataset~~ cross-dataset, new labels are defined based on whether different usernames correspond to the same author, and episodes are sampled from the corresponding markets. The overall loss function is the sum of the losses across the markets.

2.5. The Loss Function

We use the username as a label for the episode within the market and denote each username as a unique label. To train the user embedding function $f(\theta)$, we compose it with a discriminative classifier $g(\phi): RE \rightarrow RY$ with parameters ϕ to predict the author of an episode,
where Y is the number of authors in the training set. We follow the architecture proposed by Maneriker et al. (2021) [11] and use Softmax as the loss function.

3. EXPERIMENTS

3.1. Experiment Setup

Dataset and Metrics. Munksgaard and Demant (2016) [19] studied the politics of darknet markets using structured topic models on the forum posts across six large markets. Maneriker et al. (2021) [11] focus on four of the six markets – Silk Road (SR), Silk Road 2.0 (SR2), Agora Marketplace (Agora), and Black Market Reloaded (BMR) (with summary statistics in Table 2). Implementation Details. We evaluated our method using retrieval–based metrics over the representations generated by each approach. We denoted the set of all episode representations as E = {e1, e2, · · · , en} and Q = {q1, q2, · · · , qk } ∈ E is the sampled subset. We computed the cosine similarity of the query episode representations with all episodes. Let Ri =< ri1, ri2, · · · , rin > denote the list of episodes in E ordered by their cosine similarity with episode qi (excluding itself). These settings mainly ~~follows~~ follow previous work [11]. The following measures are computed. Mean Reciprocal Rank: (MRR) The RR for an episode is the reciprocal rank of the first element (by similarity) with the same author. MRR is the mean of reciprocal ranks for a sample of episodes.

Formula(8)

Recall@k: (R@k) Following Andrews and Bishop (2019) [10], we define the R@k for an episode ei to be an indicator denoting whether an episode by the same author occurs within the subset < ri1, ri2, · · · , rin >. R@k denotes the mean of these recall values over all the query samples.

3.2. Baselines

We compare our URM4DMU against three kinds of baselines. First, the Short Text Authorship Attribution Model: it ~~modeling~~ models each post with text classification models, such as TextCNN [9] and Transformer [13]. Second, we use ~~the~~ Single–Task Learning. ~~The first~~ One is IUR [10], which only considers one dataset at a time. The other one is SYSML [11], a representation learning approach ~~, which~~ that couples temporal content. Finally, we use Multi–Task Learning, a framework for training the proposed models in a multitask setting across multiple darknet markets, as described ~~at~~ in 2.4.

3.3. Main Results

Table 2 shows the results on the four datasets. The average improvements on MRR value and Recall@10 are 22.5% and 25.5% over the state–of–the–art method respectively. It shows that the MRR value of URM4DMU outperforms SYSML by 12.8% and 27.8% at most under the single–task and multiple–task settings on Agora dataset respectively. The improvement gap between the single–task setting and the multiple–task setting confirms that the transformer has stronger learning ability for a large amount of data. The improvement of BMR is not as good as the other three, since the data scale of BMR is about one–tenth of the others ~~leads this~~.

3.4. Ablation Study

We perform ablation studies to investigate the contributions of specific components of URM4DMU. Due to space limitations, we only present results on the two datasets. As described above, the graph context component models the forum interactions information, and the time component models

the temporal content information. We conduct ablation studies on the graph context component and the time component respectively. Results in Table 3 indicate that the time and the graph context are integral components ~~for~~ of ~~the~~ user representation. Besides, adding the time and ~~the~~ graph context or not, URM4DMU always outperforms SYSML.

Figure 3 shows the user alignment results with the representations of SYSML and URM4DMU. We could see that URM4DMU shows obvious advantages over CNN-based models and Transformer-based models. The reason is that URM4DMU can not only capture local n-gram features and global dependencies effectively but also preserve sequential information. It also reveals the importance and remarkable effect of combining the temporal content and forum interactions with URM4DMU for modeling darknet markets users.

4. CONCLUSION

In this paper, we propose a model named URM4DMU~~-~~, which addresses the problem of learning an invariant representation of each user based on their posts on different markets. The core contribution is the effective ~~usage~~ use of Transformer in modeling ~~a user's~~ posts whose length ~~changed~~ change in a large range frequently and, ~~especially~~ in particular, coupled with the temporal content and forum interactions of the user. The consistent and remarkable improvements on four public darknet markets datasets demonstrate the effectiveness of our URM4DMU. By linking users across multiple darknet markets, URM4DMU provides more valuable information for the analysis of anonymous users' identities ~~by linking users across multiple darknet markets~~. In the future, we will extend URM4DMU to other darknet platforms like Telegram [20] and Blockchain [21] to collect more valuable information for the analysis of anonymous users' identities. URM4DMU will also shed light on the content ~~or~~ and user analysis works related to conversation understanding[22].

Figure & Table

Fig. 1. An example with the length of a user's posts which changes in a large range frequently. The content form of darknet markets is similar to the forum.

Table3. Ablation study on URM4DMU. MRR and Recall@10 scores are reported on the test sets of SR2 and BMR datasets.